



## I. Pendahuluan

MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang software dan konsultan database bernama MYSQL AB yang berada di Swedia. Waktu itu perusahaan tersebut masih bernama TcX DataKonsult AB, dan tujuan awal dikembangkannya MySQL adalah untuk mengembangkan aplikasi berbasis web pada client. Awalnya Michael "Monty" Widenius, pengembang satu-satunya di TcX memiliki sebuah aplikasi UNIREG dan rutin ISAM buatannya sendiri dan sedang mencari antarmuka SQL yang cocok untuk diimplementasikan ke dalamnya. Mula-mula Monty memakai miniSQL (mSQL) pada eksperimennya itu, namun SQL dirasa kurang sesuai, karena terlalu lambat dalam pemrosesan query. Akhirnya Monty menghubungi David Hughes, pembuat mSQL yang sedang merilis versi kedua dari mSQL. Kemudian Monty mencoba membuat sendiri mesin SQL yang memiliki antarmuka mirip dengan SQL, tetapi dengan kemampuan yang lebih sesuai sehingga lahirlah MySQL. Tentang pengambilan nama MySQL, sampai saat ini masih belum jelas asal usulnya. Ada yang berpendapat nama My diambil dari huruf depan dan belakang Monty, tetapi versi lain mengatakan nama itu diambil dari putri Monty yang kebetulan juga bernama My.

## II. Keistimewaan MySQL

Sebagai database server yang memiliki konsep database modern, MySQL memiliki banyak sekali keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL:

### a. Portability

MySQL dapat berjalan stabil pada berbagai OS seperti Windows, Linux, Unix, Mac OS, Solaris, Unix, Amiga, HP-UX, Symbian.

### b. Open Source "limited"

Dahulu MySQL didistribusikan secara open source (gratis), dibawah lisensi GPL sehingga kita dapat menggunakannya secara cuma-cuma tanpa dipungut biaya. Namun, saat ini karena MySQL telah dibeli oleh SUN, maka kita tidak dapat lagi menikmati fitur-fitur baru yang ada di MySQL, karena SUN akan membatasi fitur-fitur baru ini hanya untuk user yang membeli lisensinya. Sehingga MySQL tidak lagi sebuah opensource yang benar-benar gratis lagi. MySQL sekarang hanya menyediakan fitur-fitur "dasar" saja yang saat ini sudah menggunakan versi 5.1. Untuk mendownloadnya silahkan download di [sini](#) dan dicari versi MySQL dengan OS kita.

### c. Multiuser

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami konflik. Hal ini memungkinkan sebuah database server MySQL dapat diakses klien secara bersamaan.

### d. Performance Tuning

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

### e. Column Types

MySQL memiliki tipe kolom yang sangat kompleks, seperti signed/unsigned integer, float,

double, char, varchar, text, blob, date, time, datetime, timestamp, year, set serta enum.

#### **f. Command dan Functions**

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.

#### **g. Security**

MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

#### **h. Scalability dan Limits**

MySQL ampu menangani database dalam skala besar dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu, batas index yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

#### **i. Connectivity**

MySQL dapat melakukan koneksi dengan klien menggunakan TCP/IP, Unix soket (Unix), atau Named Pipes (NT).

#### **j. Localisation**

MySQL dapat mendeteksi pesan kesalahan (error code) pada klien dengan menggunakan lebih dari dua puluh bahasa.

#### **k. Interface**

MySQL memiliki interface terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

#### **l. Clients dan Tools**

MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

#### **m. Struktur Tabel**

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE dibandingkan database lainnya.

### **III. Ketentuan Memberi Perintah dalam MySQL**

Berikut adalah ketentuan-ketentuan member perintah pada MySQL:

- Setiap perintah harus diakhiri dengan tanda titik koma (;) atau dengan memberikan perintah g atau G. Namun, pada umumnya user menggunakan tanda titik koma untuk mengakhiri perintah pada MySQL.
- Setiap perintah akan disimpan dalam buffer (memori sementara) untuk menyimpan histori perintah-perintah yang pernah diberikan.
- Perintah dapat berupa perintah SQL atau perintah khusus MySQL.
- Perintah-perintah yang bukan SQL dapat dipendekkan dengan menggunakan `USE` dan huruf depan perintah.
- Perintah help atau h digunakan untuk menampilkan daftar dan aturan memberikan perintah di lingkungan MySQL.
- Perintah-perintah dalam lingkungan MySQL tidak menerapkan aturan *case sensitive*, tetapi

#### *case insensitive*

yaitu perintah bisa dituliskan dalam huruf besar atau pun huruf kecil.

- Aturan *case sensitive* diterapkan pada penamaan objek-objek dalam database seperti nama database atau nama table, namun aturan ini hanya ada dalam lingkungan Unix dan Linux.
- Untuk melihat perintah-perintah yang sudah pernah kita ketikkan, tekan tombol tanda panah atas.

## IV. Perintah-Perintah dasar MySQL

Untuk mengikuti tutorial ini, dianggap MySQL sudah terinstal dengan baik dalam sistem. Jika menggunakan Linux Centos, dan terhubung dengan internet, maka untuk menginstal MySQL cukup ketikkan:

# yum -y install mysql\* maka MySQL akan langsung terinstal ke dalam sistem Linux secara otomatis. Setelah itu nyalakan MySQL dengan cara:

# service mysqld start maka MySQL siap untuk digunakan. Penulis menggunakan Linux Centos 5 dan MySQL versi 5.0 di dalam tutorial ini. Tulisan yang diberikan cetak tebal, maka itu berarti perintah-perintah MySQL.

### 1. Mengecek Paket MySQL

Untuk mengecek paket-paket MySQL, ketikkan perintah:

# rpm -qa | grep mysql

### 2. Melihat versi MySQL

Untuk melihat versi MySQL yang ada dalam sistem, ketikkan:

# mysql --version

### 3. Masuk ke MySQL

Ketikkan mysql pada sistem agar kita bisa masuk ke dalam database MySQL.

```
[root@Centos5 ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 243
Server version: 5.0.45 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

### 4. Melihat Database

Untuk melihat database apa yang di MySQL, ketikkan:

> **show databases;**

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cacti |
| joomla |
| mediawiki |
| mysql |
| postfix |
| test |
| wordpress |
| zabbix |
+-----+
9 rows in set (0.01 sec)
```

### 5. Membuat Database baru

Untuk membuat database baru, ketikkan perintah:

> **create database** nama\_database; Ingat: nama\_database dirubah menjadi nama database sesuai dengan anda. Dalam tutorial ini kita akan membuat database latihan, sehingga perintahnya menjadi:

> **create database** latihan;

### 6. Menggunakan Database;

Untuk menggunakan database yang ada di MySQL, ketikkan perintah:

> **use** nama\_database; Karena kita telah membuat database latihan, maka kita akan menggunakan database ini, sehingga perintah tersebut menjadi:

> **use** latihan;

### 7. Membuat Table

Di dalam database latihan, kita akan membuat tabel-tabel dengan format:

> **create table** nama\_tabel (nama\_data tipe\_data (panjang\_karakter), ...); Tipe-tipe data pada MySQL bermacam-macam dan dapat dilihat lengkap di sini. panjang\_karakter merupakan jumlah karakter maksimum yang ditampung di dalam sebuah field. Kali ini, kita akan membuat tabel alamat yang berisi field no, nama, alamat, kota, telepon.

```
mysql> create table alamat
-> (
-> no int(3),
-> nama char(25),
-> alamat varchar(40),
-> kota char(15),
-> usia int(3)
-> );
Query OK. 0 rows affected (0.01 sec)
```

### 8. Melihat Table

Untuk melihat table yang sudah kita buat, ketikkan perintah:

> **show tables**; Namun, untuk melihat sebuah table yang lebih lengkap beserta tipe-tipe datanya, ikuti format berikut:

> **describe** nama\_table; Dalam kasus kita,

> **describe** alamat;

```
mysql> show tables;
+-----+
| Tables_in_latihan |
+-----+
| alamat             |
+-----+
1 row in set (0.00 sec)

mysql> describe alamat;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| no     | int(3)        | YES  |     | NULL    |       |
| nama   | char(25)      | YES  |     | NULL    |       |
| alamat | varchar(40)   | YES  |     | NULL    |       |
| kota   | char(15)      | YES  |     | NULL    |       |
| usia   | int(3)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

## 9. Merubah Nama Field

Apabila kita ingin merubah nama field pada tabel yang sudah jadi, maka harus mengikuti format:

> **alter table** nama\_table **change** nama\_field\_lama nama\_field\_baru tipe\_data(panjang\_karakter); Pada perintah ini, tipe\_data dan panjang\_karakter dapat dirubah. Dalam kasus kita:

> **alter table** alamat **change** no nomor varchar(10); Untuk melihatnya apa yang telah kita rubah, ketikkan:

> **describe** alamat;

## 10. Merubah Tipe Data

Jika kita ingin merubah tipe data sebuah field, maka harus mengikuti format:

> **alter table** nama\_table **modify** nama\_field tipe\_data\_baru(panjang\_karakter); Dalam kasus kita:

> **alter table** alamat **modify** kota varchar(17);

## 11. Menambah Kolom/Field

Format untuk menambah kolom/field pada sebuah tabel adalah:

> **alter table** nama\_table **add** field\_baru tipe\_data(panjang\_karakter); Dalam kasus kita:

> **alter table** alamat **add** email varchar(20);

## 12. Menghapus Field/Kolom

Format untuk menghapus kolom/field pada sebuah tabel adalah:

> **alter table** nama\_table **drop** nama\_field; Dalam kasus kita:

> **alter table** alamat **drop** email;

## 13. Mengganti Nama Tabel

Format untuk mengganti nama tabel adalah:

- > **alter table** nama\_table\_lama **rename** nama\_table\_baru; Dalam kasus kita:
- > **alter table** alamat **rename** teman;

#### 14. Memasukkan Data-Data

Format untuk memasukkan data-data pada sebuah tabel adalah:

- > **insert into** nama\_table **values** ('data\_field\_1','data\_field\_2',...); Dalam kasus kita:

```
mysql> insert into teman values (1,'agus suga','jl sawo no 7','jakarta',23);
Query OK, 1 row affected (0.00 sec)

mysql> insert into teman values (2,'toto otot','jl kangga no 9','jakarta',24);
Query OK, 1 row affected (0.00 sec)

mysql> insert into teman values (3,'yuyu uyuy','jl jabu no 67','bandung',24);
Query OK, 1 row affected (0.00 sec)

mysql> insert into teman values (4,'ivan navi','jl bangkono no 34','surabaya',27);
Query OK, 1 row affected (0.01 sec)

mysql> insert into teman values (5,'toto otot','jl belimbing no 34','semarang',25);
Query OK, 1 row affected (0.00 sec)
```

#### 14. Mengubah Data-Data

Format untuk merubah data-data pada sebuah tabel adalah:

- > **update** nama\_table **set** nama\_field='isian\_baru' **where** nama\_field\_lain='isi\_field\_tersebut';

Contoh:

- > **update** teman **set** nama='hari irah' **where** no=2; maka nama seseorang yang berada di nomor 2 akan berubah menjadi hari irah.

#### 15. Melihat Data-Data

Format umum untuk melihat data-data pada sebuah tabel adalah:

- > **select** nama\_field **from** nama\_table **where** nama\_field\_lain='isi\_field\_tersebut'; Agar bisa lebih mengerti dengan pembahasan ini, cobalah tambah beberapa data seperti pada poin 13 dengan beberapa field memiliki nilai yang sama. Karena untuk melihat table memiliki banyak format, maka kita bisa melihat sebuah tabel dengan berbagai macam cara, seperti:

##### a. Menampilkan semua kolom

Format:

- > **select \* from** nama\_table; Contoh:
- > **select \* from** teman;

```
mysql> select * from teman;
```

no	nama	alamat	kota	usia
1	agus suga	jl sawo no 7	jakarta	23
2	toto otot	jl kangga no 9	jakarta	24
3	yuyu uyuy	jl jabu no 67	bandung	24
4	ivan navi	jl bangkono no 34	surabaya	27
5	toto otot	jl belimbing no 34	semarang	25

5 rows in set (0.01 sec)

##### b. Menampilkan kolom tertentu

Format:

- > **select** nama\_field **from** nama\_table; Contoh:
- > **select** nama, telepon **from** alamat;

```
mysql> select nama,alamat from teman;
```

nama	alamat
agus suga	jl sawo no 7
hari irah	jl mangga no 9
juju ujuj	jl jambu no 67
iwan nawi	jl bangkoang no 34
toto otot	jl belimbing no 34

```
5 rows in set (0.00 sec)
```

#### c. Menampilkan kolom tertentu dengan kondisi tertentu

Format:

> **select** nama\_field **from** nama\_table **where** nama\_field\_lain='isi\_field\_tersebut'; Contoh:

> **select** nama **from** teman **where** kota='jakarta';

```
mysql> select nama from teman where kota='jakarta';
```

nama
agus suga
hari irah

```
2 rows in set (0.00 sec)
```

#### d. Menghindari Pengulangan Data Pada Kolom

Format:

> **select distinct** nama\_field **from** nama\_table; Contoh:

> **select distinct** kota **from** teman;

```
mysql> select distinct kota from teman;
```

kota
jakarta
bandung
surabaya
semarang

```
4 rows in set (0.01 sec)
```

#### e. Membatasi data yang ditampilkan

Format:

> **select** nama\_field\_yang\_mau\_ditampilkan **from** nama\_table **limit**

dari\_baris,sampai\_baris;

Contoh kita akan menampilkan data yang dimulai dari baris kedua sampai keenam, maka perintahnya adalah:

> **select \* from teman limit 2,6;**

```
mysql> select * from teman limit 0,3;
+----+-----+-----+-----+
| no | nama   | alamat          | kota   |
+----+-----+-----+-----+
| 1  | agus suga | jl sawo no.7    | jakarta |
| 2  | hari irah | jl mangga no.9  | jakarta |
| 3  | juju ujuj | jl jambu no.67  | bandung |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
```

#### f. Mengurutkan data

Format:

> **select** nama\_field\_yang\_mau\_ditampilkan **from** nama\_table **order by** nama\_field **klausa**;

Klausa format ini ada 2 yaitu asc yaitu akan mengurutkan data naik dari yang terkecil dan desc yaitu kebalikannya.

```
mysql> select * from teman order by usia asc;
+----+-----+-----+-----+-----+
| no | nama   | alamat          | kota   | usia |
+----+-----+-----+-----+-----+
| 1  | agus suga | jl sawo no.7    | jakarta | 23  |
| 2  | tono omot | jl mangga no.9  | jakarta | 24  |
| 3  | juju ujuj | jl jambu no.67  | bandung | 24  |
| 4  | tono omot | jl belimbing no.34 | semarang | 25  |
| 5  | iwan nawi | jl bangkoang no.34 | surabaya | 27  |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from teman order by usia desc;
+----+-----+-----+-----+-----+
| no | nama   | alamat          | kota   | usia |
+----+-----+-----+-----+-----+
| 5  | iwan nawi | jl bangkoang no.34 | surabaya | 27  |
| 4  | tono omot | jl belimbing no.34 | semarang | 25  |
| 3  | juju ujuj | jl jambu no.67  | bandung | 24  |
| 2  | tono omot | jl mangga no.9  | jakarta | 24  |
| 1  | agus suga | jl sawo no.7    | jakarta | 23  |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

#### g. Menggabungkan beberapa kondisi

Format:

> **select** nama\_field\_yang\_mau\_ditampilkan **from** nama\_table **where** nama\_field='isi\_field'

**operator\_logika**

nama\_field='isi\_field';

Opearator\_logika disini ada 2 yaitu

**and**

**dan**

**or**

. Untuk melihat perbedaan diantara keduanya, perhatikan gambar dibawah ini:

```
mysql> select * from teman where kota='jakarta' and usia>23;
+----+-----+-----+-----+-----+
| no | nama   | alamat          | kota   | usia |
+----+-----+-----+-----+-----+
| 2  | tono omot | jl mangga no.9  | jakarta | 24  |
+----+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> select * from teman where kota='jakarta' or usia>23;
+----+-----+-----+-----+-----+
| no | nama   | alamat          | kota   | usia |
+----+-----+-----+-----+-----+
| 1  | agus suga | jl sawo no.7    | jakarta | 23  |
| 2  | tono omot | jl mangga no.9  | jakarta | 24  |
| 3  | juju ujuj | jl jambu no.67  | bandung | 24  |
| 4  | iwan nawi | jl bangkoang no.34 | surabaya | 27  |
| 5  | tono omot | jl belimbing no.34 | semarang | 25  |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

#### h. Memberi batas data

Format:

> **select** nama\_field\_yang\_mau\_ditampilkan **from** nama\_table **where** nama\_field **between** is



i\_field

and

isi\_field;

```
mysql> select * from teman where usia between 24 and 25;
```

no	nama	alamat	kota	usia
2	tono onot	jl mangga no 9	jakarta	24
3	juju ujuj	jl jambu no 67	bandung	24
5	toto otot	jl belimbing no 34	semarang	25

3 rows in set (0.00 sec)

### i. Mencari Data Dengan Kriteria Tertentu

Format:

> **select** nama\_field\_yang\_mau\_ditampilkan **from** nama\_table **where** nama\_field **like** '%';

'%' ada 3 macam, yaitu:

'keyword%': Mencari data yang diawali dengan keyword

'%keyword': Mencari data yang diakhiri dengan keyword

'%keyword%': Mencari data yang mengandung keyword

```
mysql> select * from teman where kota like 'S%';
```

no	nama	alamat	kota	usia
4	Iwan nawi	jl bangkawang no 34	surabaya	27
5	toto otot	jl belimbing no 34	semarang	25

2 rows in set (0.01 sec)

```
mysql> select * from teman where kota like 'Ja%';
```

no	nama	alamat	kota	usia
1	agus suga	jl sawo no 7	jakarta	23
2	tono onot	jl mangga no 9	jakarta	24
4	Iwan nawi	jl bangkawang no 34	surabaya	27

3 rows in set (0.00 sec)

```
mysql> select * from teman where kota like '%J%';
```

no	nama	alamat	kota	usia
3	juju ujuj	jl jambu no 67	bandung	24
4	Iwan nawi	jl bangkawang no 34	surabaya	27

2 rows in set (0.01 sec)

### j. Menghitung Sejumlah nilai (Fungsi Agregat)

Nama Fungsi

Deskripsi

SUM()

Menghitung jumlah nilai pada suatu kolom

AVG()

Menghitung nilai rata-rata pada suatu kolom

MAX()

Mencari nilai maksimal pada suatu kolom

MIN()

Mencari nilai minimal pada suatu kolom

COUNT()

Menghitung jumlah baris pada suatu kolom

Ditulis oleh administrator

Senin, 12 Januari 2009 18:19 - Terakhir Diperbaharui Rabu, 20 Mei 2009 03:57

---

```
mysql> select sum(usia) from teman;
+-----+
| sum(usia) |
+-----+
|      123 |
+-----+
1 row in set (0.25 sec)

mysql> select avg(usia) from teman;
+-----+
| avg(usia) |
+-----+
|  24.6000 |
+-----+
1 row in set (0.00 sec)

mysql> select max(usia) from teman;
+-----+
| max(usia) |
+-----+
|      27 |
+-----+
1 row in set (0.00 sec)

mysql> select min(usia) from teman;
+-----+
| min(usia) |
+-----+
|      23 |
+-----+
1 row in set (0.00 sec)

mysql> select count(*) from teman;
+-----+
| count(*) |
+-----+
|         5 |
+-----+
1 row in set (0.00 sec)
```

## 16. Menghapus Tabel

Format:

> **drop table** nama\_table; Dalam kasus kita:

> **drop table** alamat; **17. Menghapus Database**

Format:

> **drop database** nama\_database; Dalam kasus kita:

> **drop database** latihan; **18. Keluar dari MySQL**

Format: > quit; atau > exit; **Referensi:**

Belajar Sendiri Administrasi Database Server MySQL, Didik Dwi Prasetyo

<http://dev.mysql.com>

<http://infokomtek.com>

17 Muharram 1430 H

14 Januari 2009 M